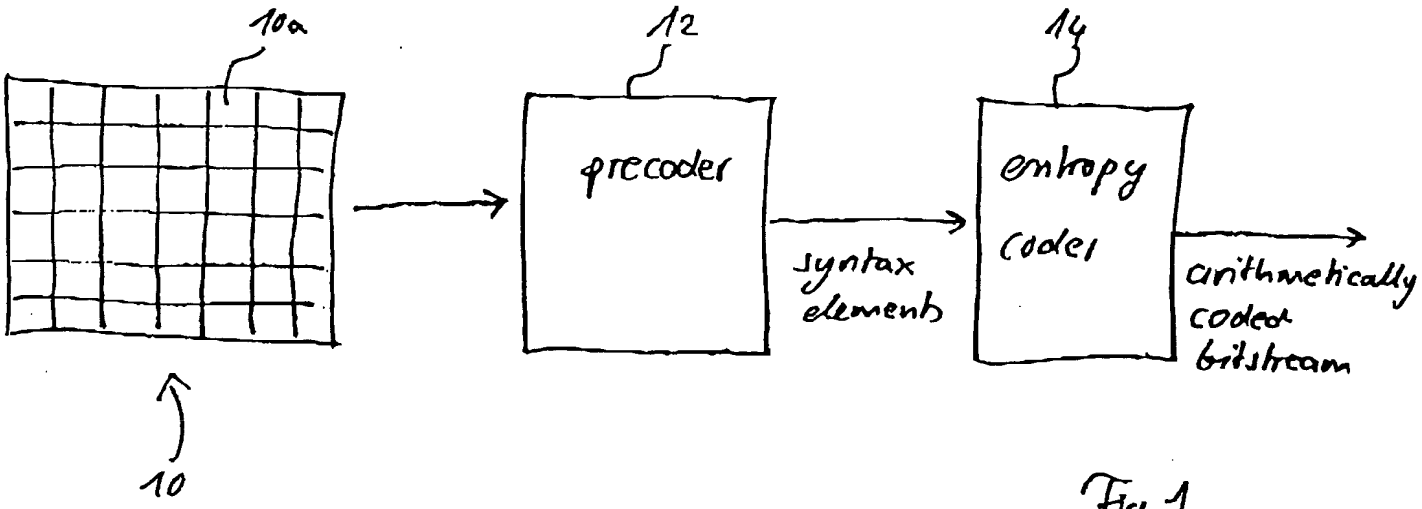

10a
12
14
precoder
entropy
coder
syntax
elements
arithmetically
coded
bitstream
10

Fig. 1

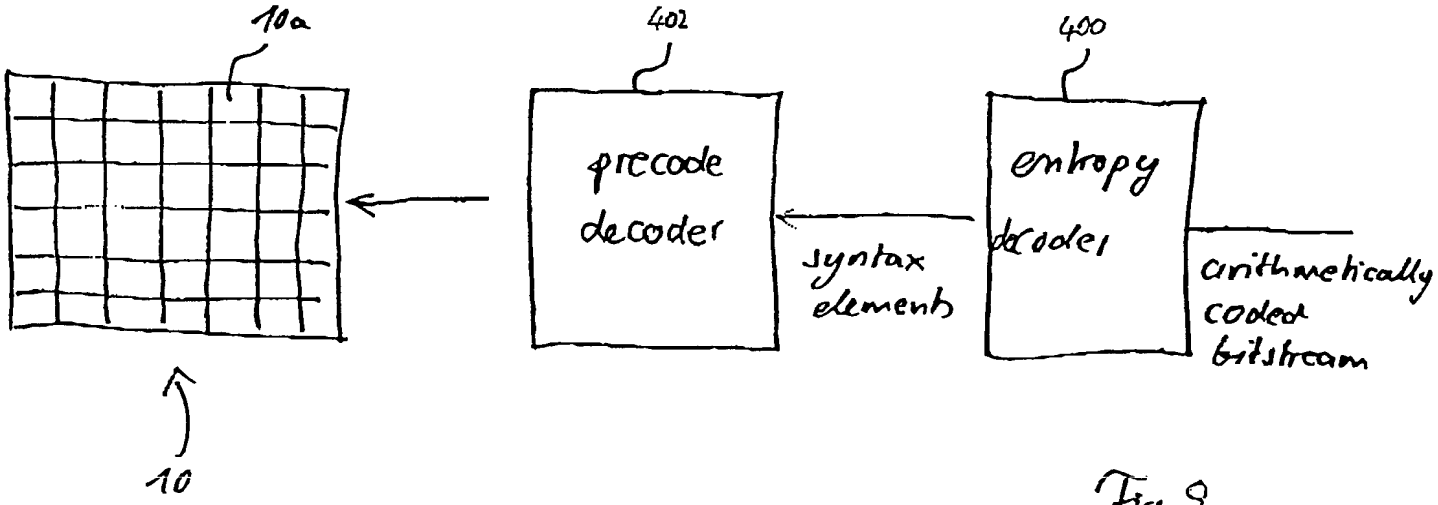

10a
402
400
precode
decoder
entropy
decoder
syntax
elements
arithmetically
coded
bitstream
10

Fig. 9

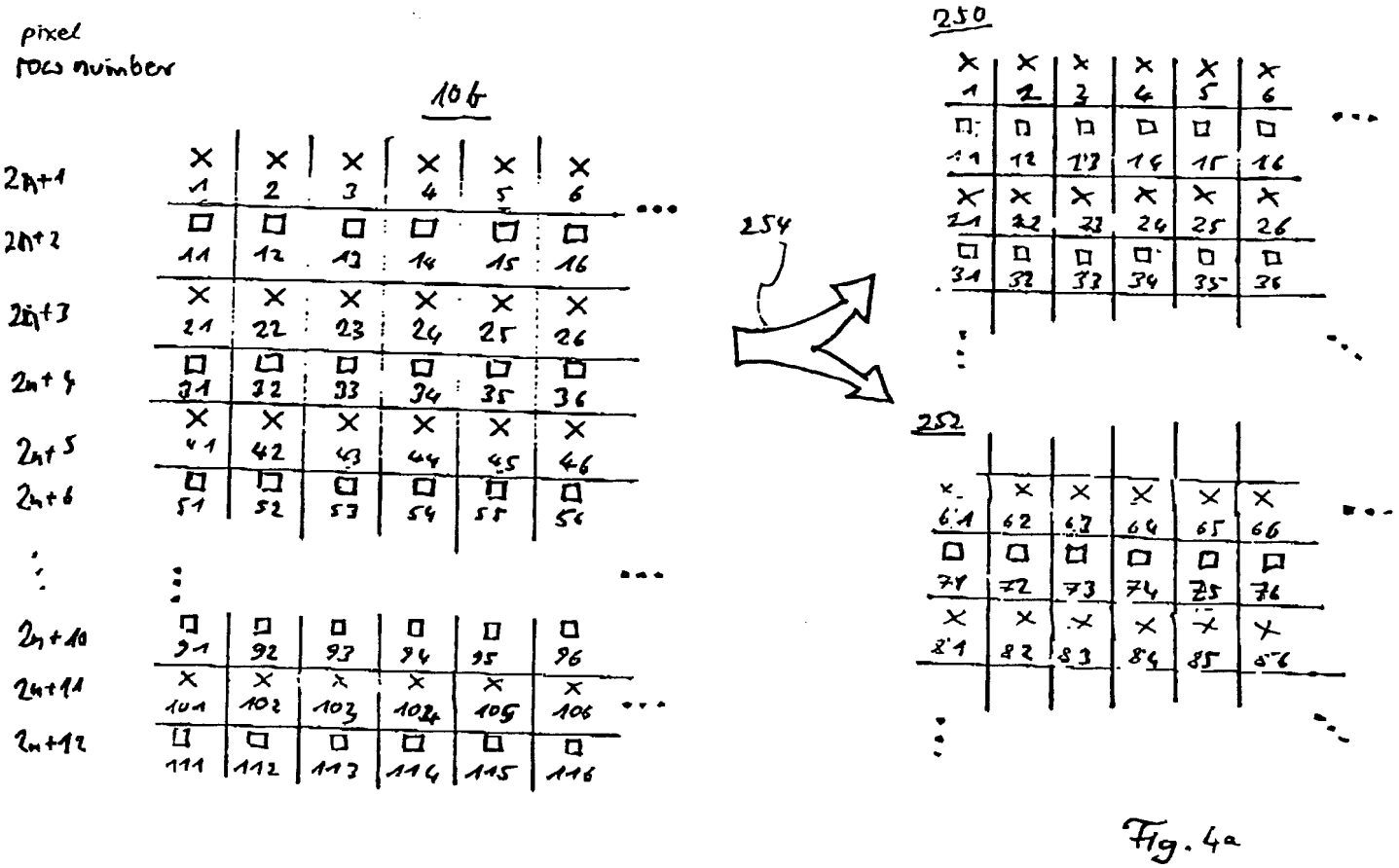

pixel
rows number
106
250
254
252

Fig. 4a

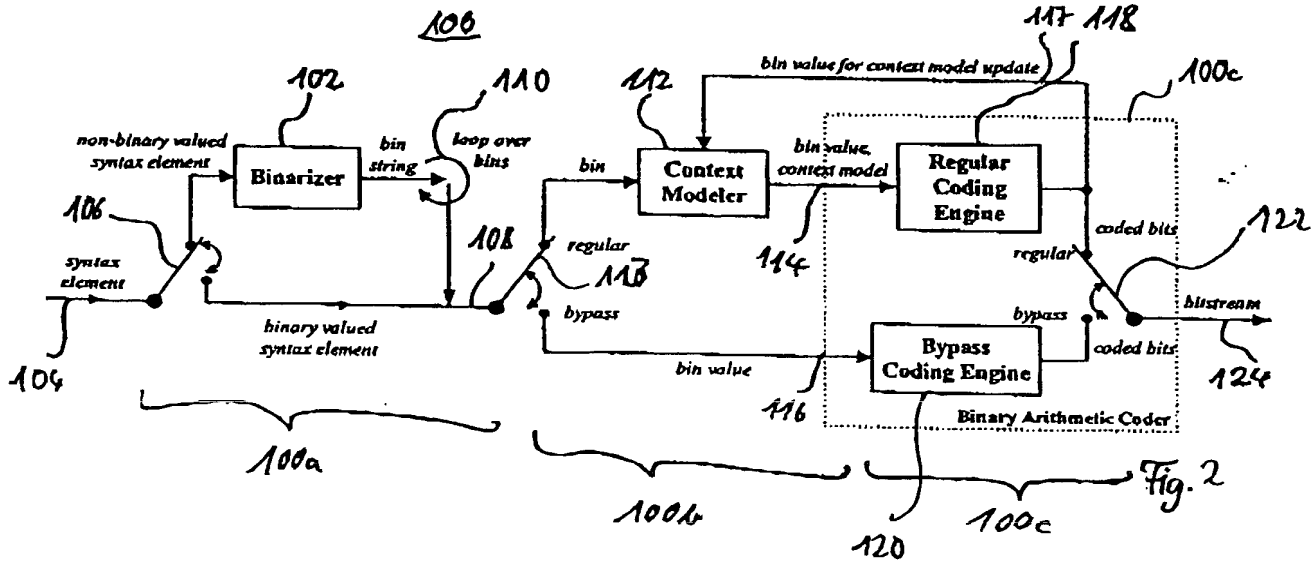

100
102
110
112
117
118
100c
non-binary valued syntax element
Binarizer
bin string
loop over bins
bin
Context Modeler
bin value for context model update
bin value, context model
Regular Coding Engine
106
syntax element
108
regular
113
114
coded bits
122
bypass
bitstream
binary valued syntax element
Bypass Coding Engine
bin value
116
Binary Arithmetic Coder
104
100a
100b
120
100c

Fig. 2

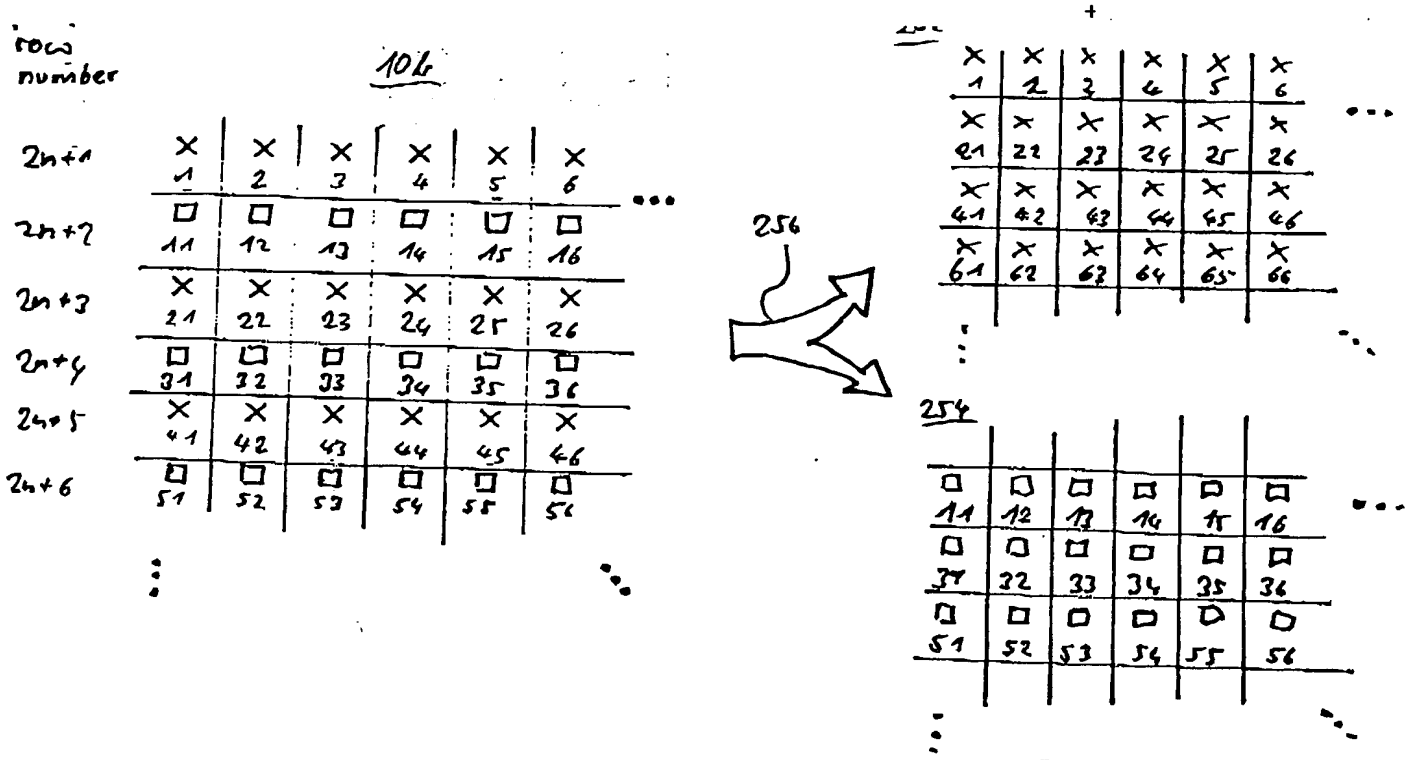


Fig. 4b

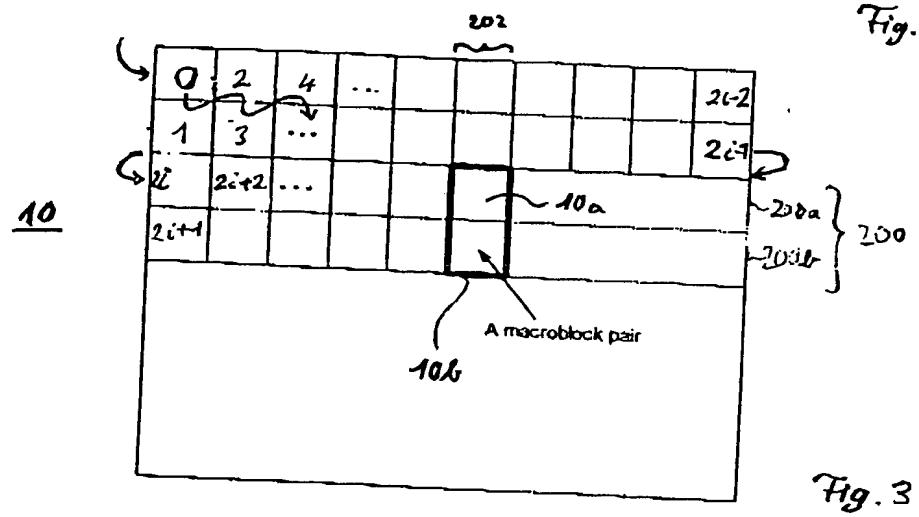


Fig. 3

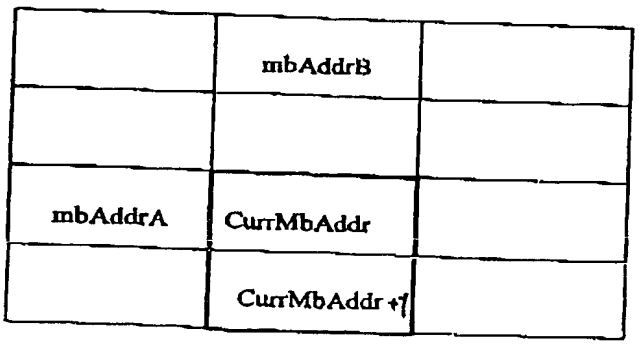


Fig. 7

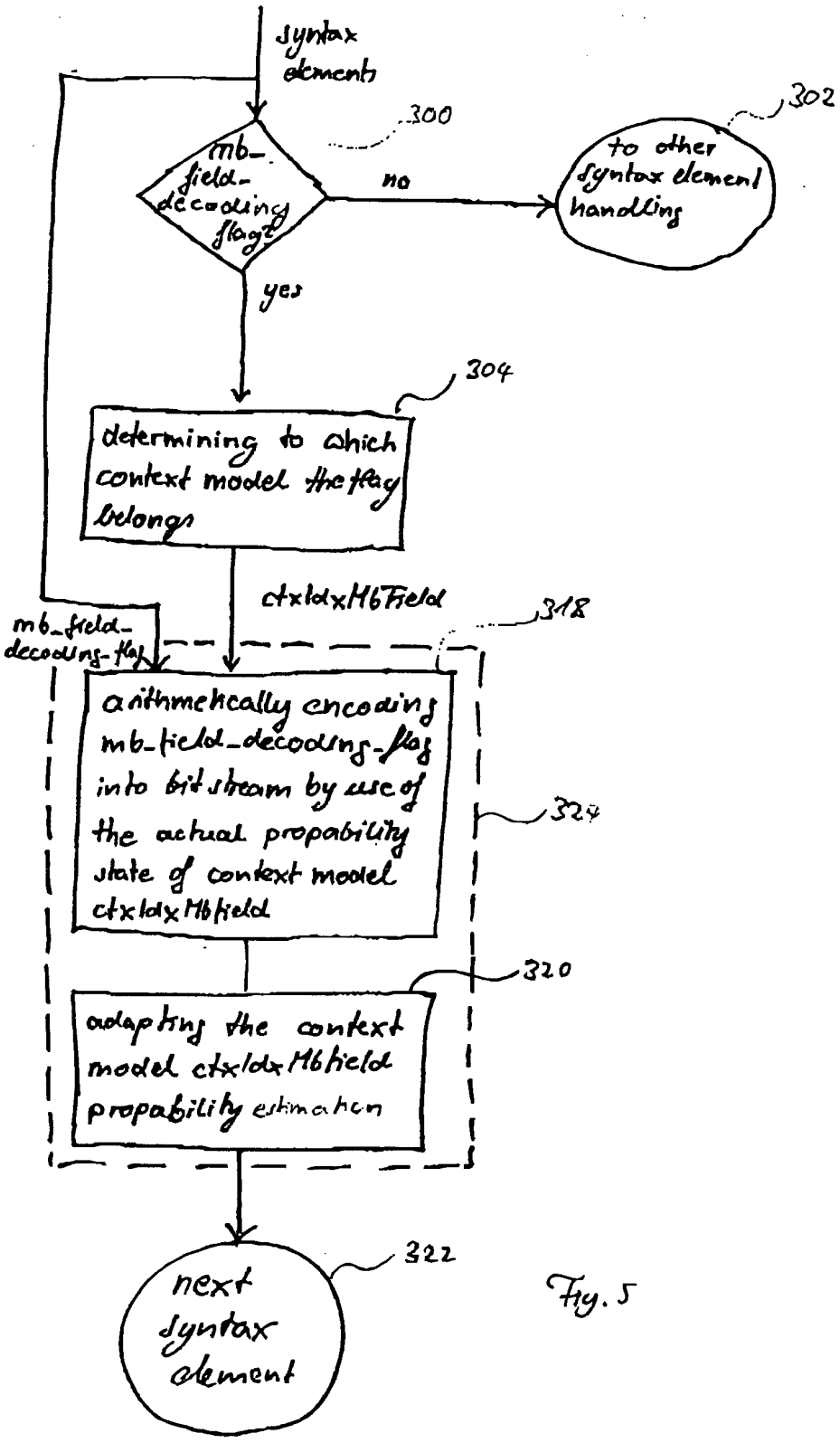

syntax elements
300
mb_field_decoding flag?
no
302
to other syntax element handling
yes
304
determining to which context model the flag belongs
ctxIdxMbField
318
mb_field_decoding_flag
arithmetically encoding mb_field_decoding_flag into bit stream by use of the actual propability state of context model ctxIdxMbfield
324
320
adapting the context model ctxIdxMbfield propability estimation
322
next syntax element

Fig. 5

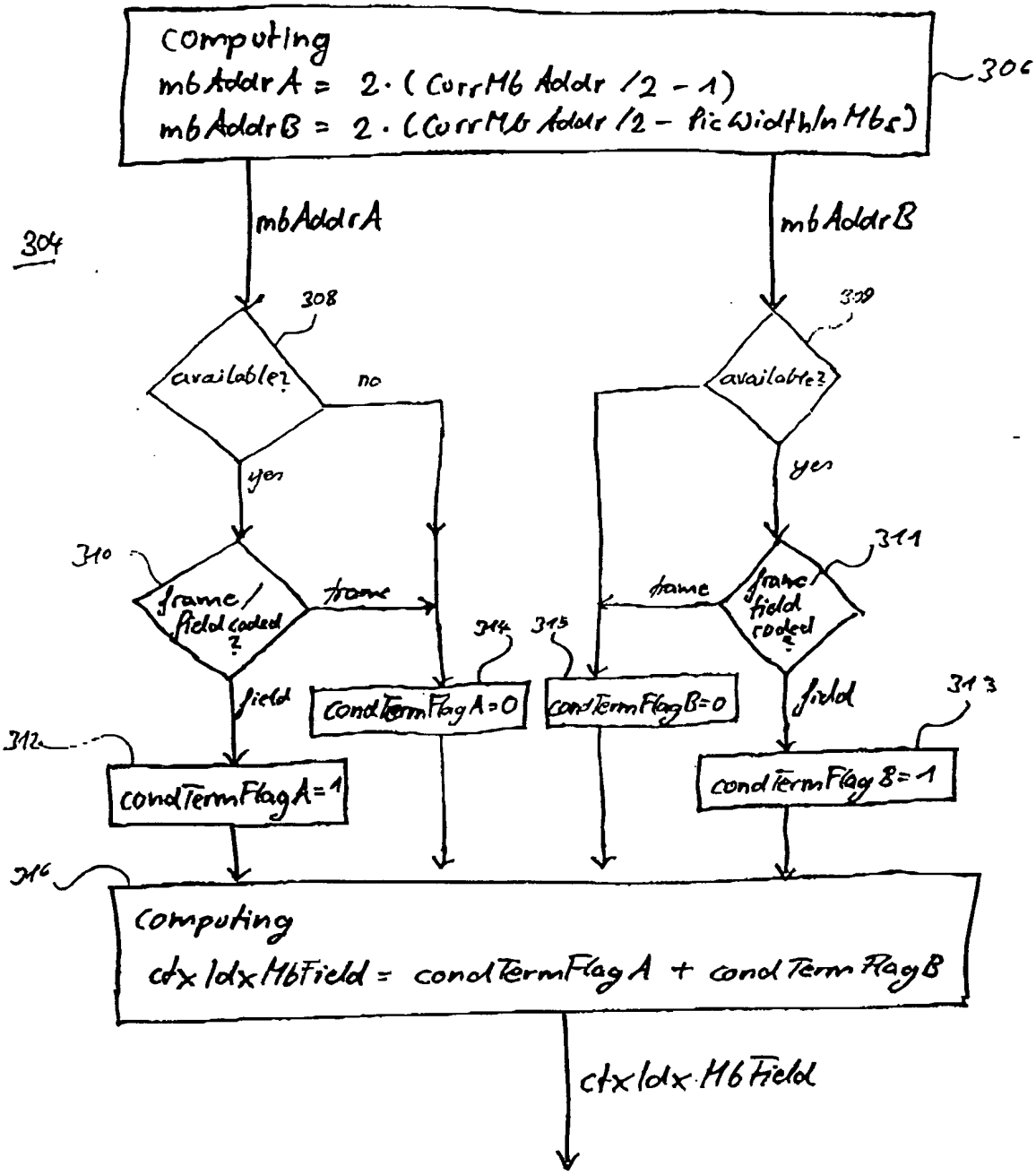

computing
mbAddrA = 2 · (CurrMb Addr /2 - 1)
mbAddrB = 2 · (CurrMb Addr /2 - PicWidthInMbs)
306
304
mbAddrA
mbAddrB
308
309
available?
no
yes
available?
yes
310
311
frame/field coded ?
frame
frame
frame/field coded ?
314
315
condTermFlagA=0
condTermFlagB=0
field
field
313
312
condTermFlagA=1
condTermFlagB=1
316
computing
ctxIdxMbField = condTermFlagA + condTermFlagB
ctxIdx MbField

Fig. 6

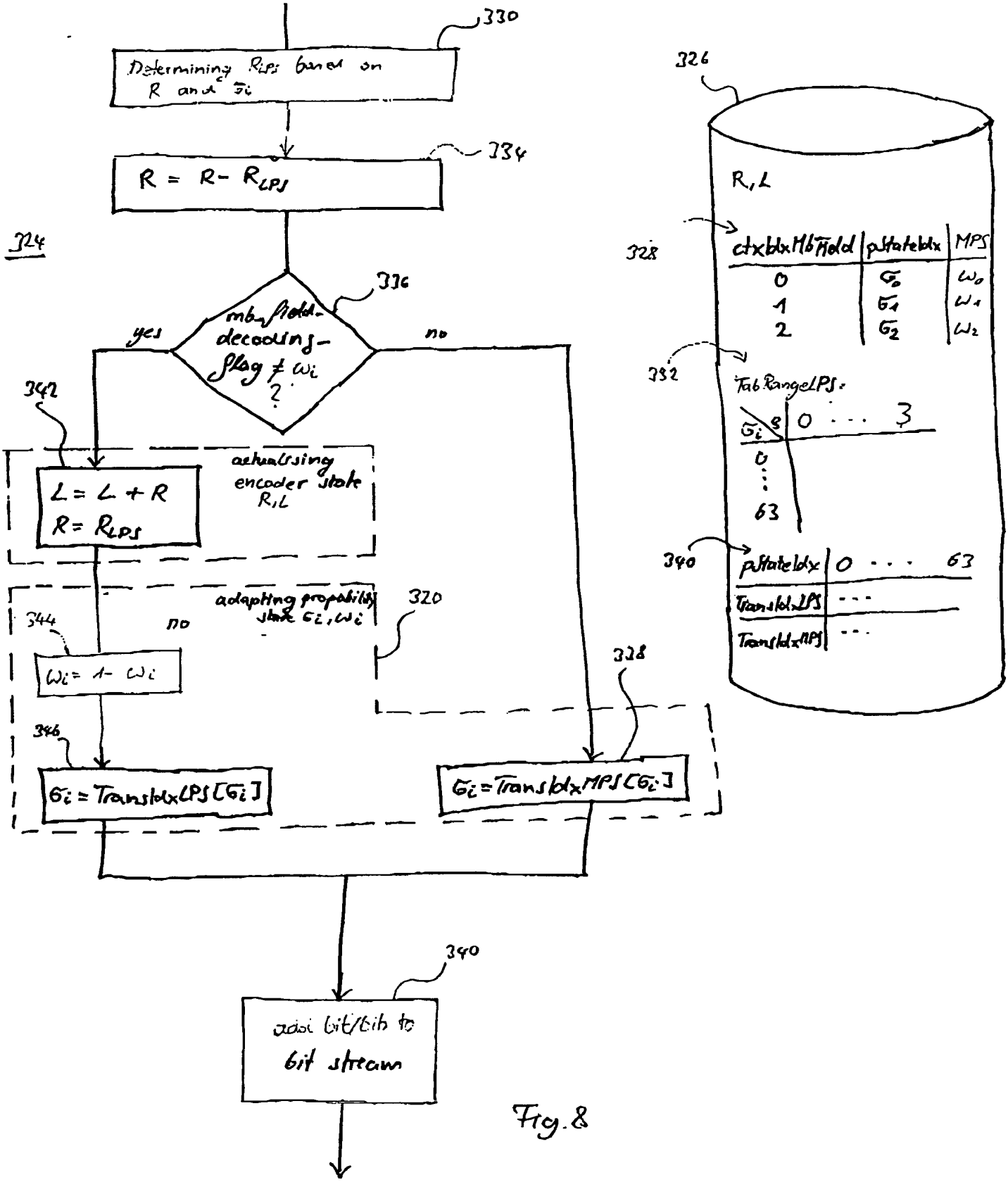

330
Determining R_{LPS} based on R and σ_i
334
$R = R - R_{LPS}$
324
336
mb_field_decoding_flag $\neq \omega_i$?
yes
no
342
$L = L + R$
$R = R_{LPS}$
actualising encoder state R,L
adapting probability state σ_i, ω_i
320
344
no
$\omega_i = 1 - \omega_i$
346
$\sigma_i = TransIdxLPS[\sigma_i]$
338
$\sigma_i = TransIdxMPS[\sigma_i]$
340
add bit/bits to bit stream
326
R, L
328
ctxIdx MbField | pStateIdx | MPS
0 | σ_0 | ω_0
1 | σ_1 | ω_1
2 | σ_2 | ω_2
332
TabRangeLPS:
σ_i \ q | 0 ... 3
0
63
340
pStateIdx | 0 ... 63
TransIdxLPS | ...
TransIdxMPS | ...

Fig. 8

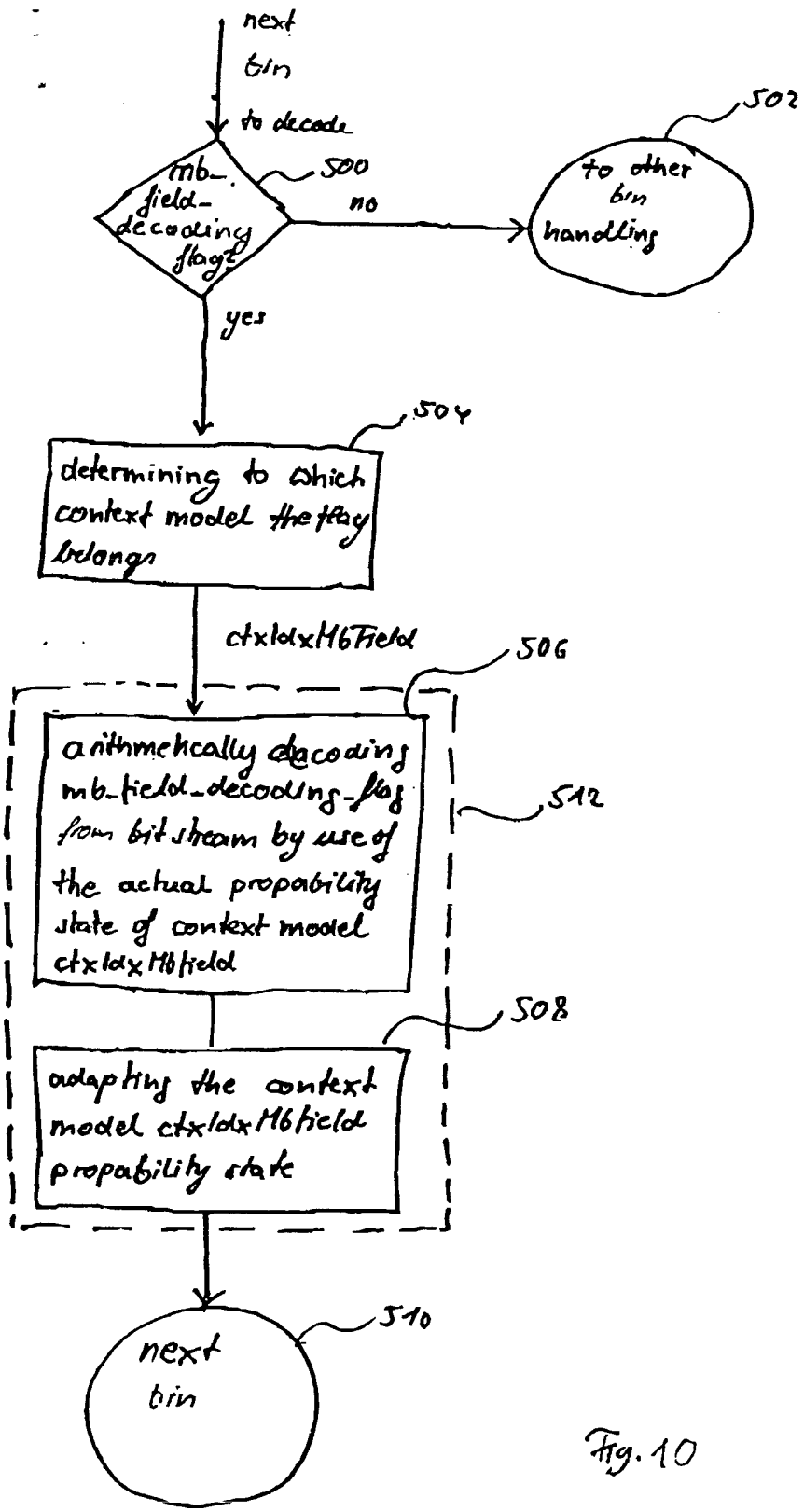

next
bin
to decode
mb_
field_
decoding
flag?
500
no
502
to other
bin
handling
yes
504
determining to which
context model the flag
belongs
ctxIdxMbField
506
arithmetically decoding
mb_field_decoding_flag
from bit stream by use of
the actual propability
state of context model
ctxIdxMbfield
512
508
adapting the context
model ctxIdxMbfield
propability state
510
next
bin

Fig. 10

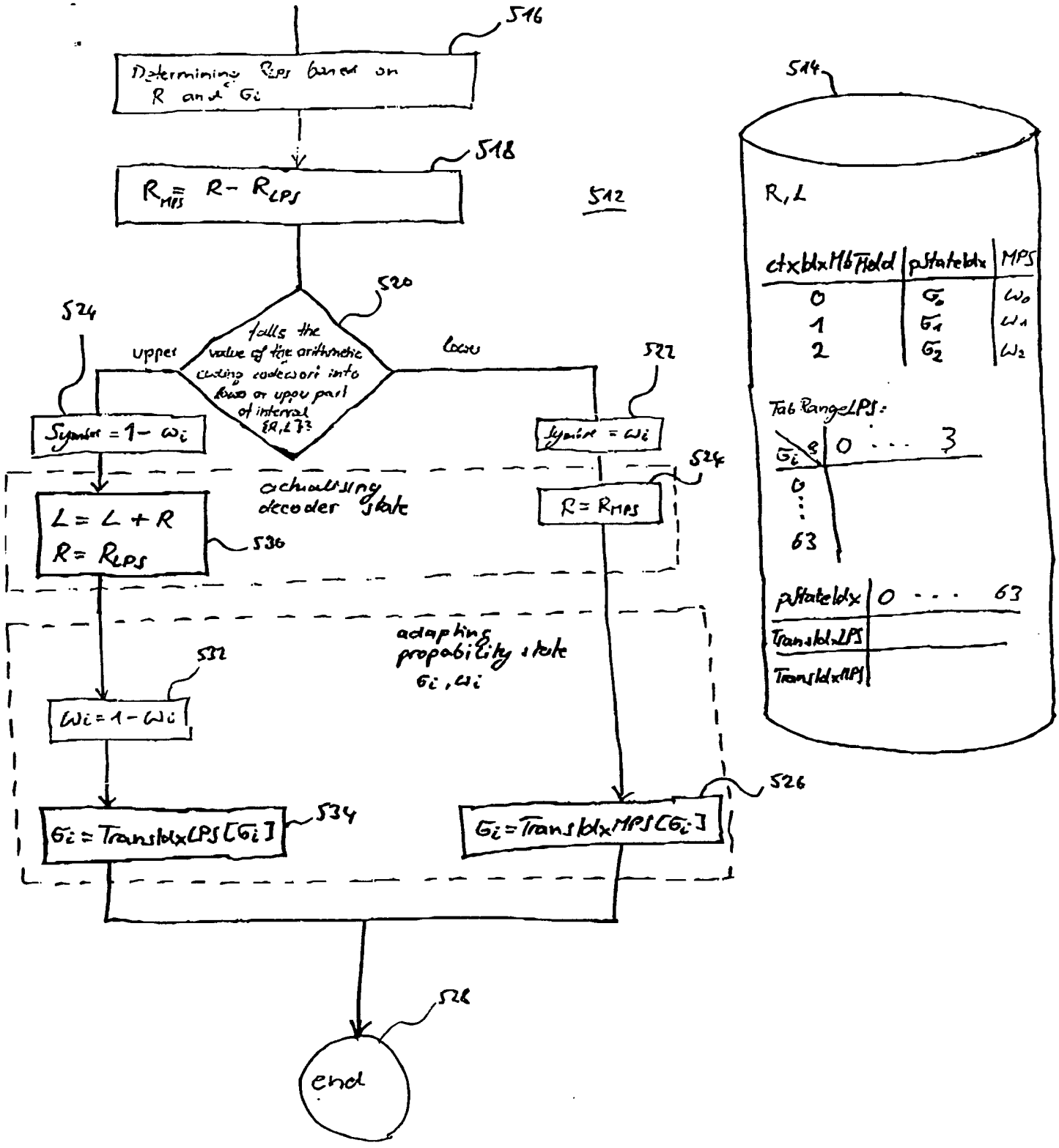

516
Determining RLPS based on R and Gi
518
RMPS = R - RLPS
512
514
R, L
ctxIdx MbField | pStateIdx | MPS
0 | G0 | ω0
1 | G1 | ω1
2 | G2 | ω2
Tab RangeLPS:
Gi | Q: 0 ... 3
0
63
pStateIdx | 0 ... 63
TransIdxLPS
TransIdxMPS
520
falls the value of the arithmetic coding codeword into lower or upper part of interval [R,L]?
upper
lower
524
522
Symbol = 1 - ωi
Symbol = ωi
actualizing decoder state
L = L + R
R = RLPS
536
R = RMPS
524
adapting probability state Gi, ωi
532
ωi = 1 - ωi
Gi = TransIdxLPS[Gi]
534
Gi = TransIdxMPS[Gi]
526
end
528

Fig. 11

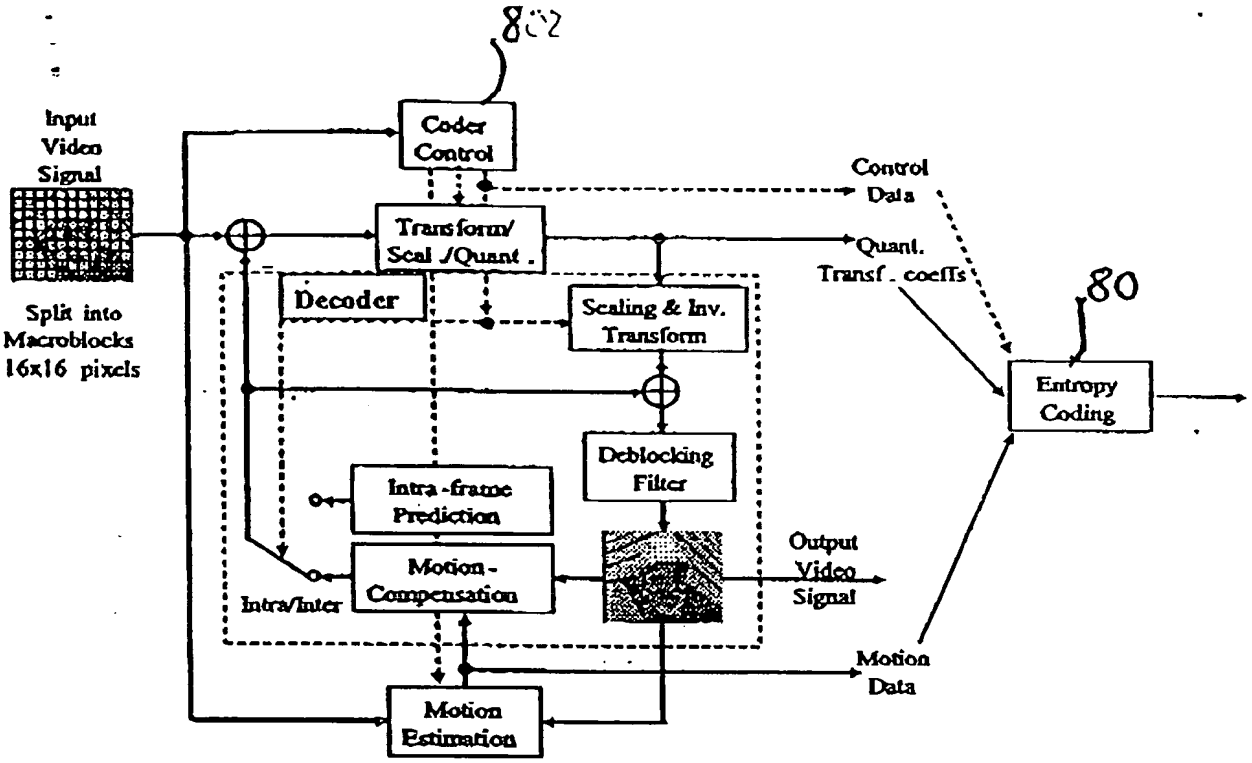
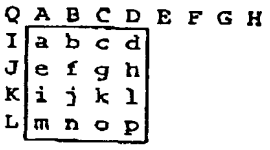


Fig. 12



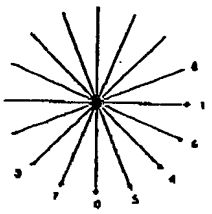


Fig. 15

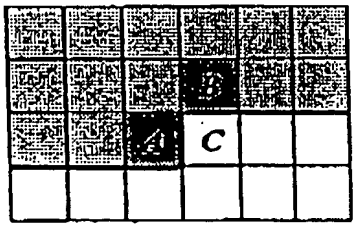


Fig. 13

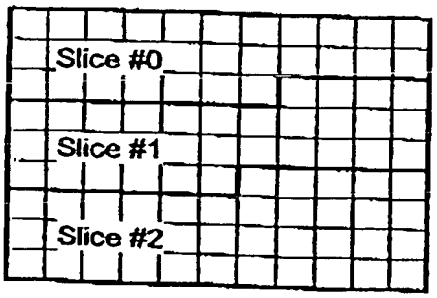


Fig. 14